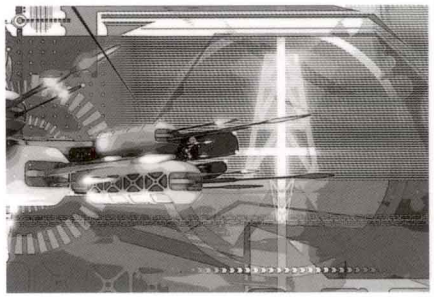
工业和信息化人才培养规划教材
高职高专计算机系列

C语言程序设计教程（第3版）



The C Programming Language

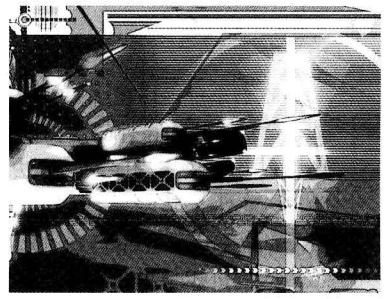
宗大华 陈吉人 宗涛 ◎ 编

本书以严谨而通俗的语言，丰富的示例，讲述C语言的语法和编程技术，使初学者能够建立起正确的概念，掌握语言本身的特征，学会基本的编程方法，形成对C语言的整体了解。

人民邮电出版社
POSTS & TELECOM PRESS
精品系列

工业和信息化人才培养规划教材　高职高专计算机系列

C语言程序设计教程（第3版）



宗大华 陈吉人 宗涛 ◎ 编

人民邮电出版社
北京

图书在版编目（CIP）数据

C语言程序设计教程 / 宗大华，陈吉人，宗涛编. -- 3版. -- 北京 : 人民邮电出版社，2012.9
工业和信息化人才培养规划教材. 高职高专计算机系列
ISBN 978-7-115-28928-5

Ⅰ. ①C… Ⅱ. ①宗… ②陈… ③宗… Ⅲ. ①C语言－程序设计－高等职业教育－教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2012)第174872号

内 容 提 要

本书系统地讲述了C语言程序设计的基本知识和方法，内容分为9章：概述，数据类型、运算符与表达式，C语言程序设计的3种基本结构，数组，指针，函数，用户自定义的数据类型，C语言程序的文件操作函数以及C语言程序调试方法简介。本书力求使学生在学习的基础上，掌握编程和调试程序的基本技术。除第9章外，其余每章最后配有适量的练习题供教学使用。在人民邮电出版社的教学服务与资源网（www.ptpedu.com.cn）上，读者可以得到有关本书的电子教案和习题参考答案。

本书可作为高职高专计算机及相关专业的教材，也可作为成人教育和职工培训教材。

工业和信息化人才培养规划教材——高职高专计算机系列

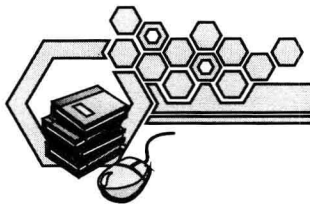
C语言程序设计教程（第3版）

◆ 编 宗大华 陈吉人 宗 涛
责任编辑 桑 珊
◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
北京昌平百善印刷厂印刷
◆ 开本：787×1092 1/16
印张：17.25 2012年9月第3版
字数：441千字 2012年9月北京第1次印刷

ISBN 978-7-115-28928-5

定价：36.00元

读者服务热线：(010)67170985 印装质量热线：(010)67129223
反盗版热线：(010)67171154



前　言

作为一种程序设计语言，C 语言语句简洁、使用灵活、性能高效、应用面广，有着极其独特的魅力，已成为计算机及相关专业人员学习计算机程序设计的首选。

本书是一本为高职高专学生编写的 C 语言教材。本书针对读者的特点和认知能力，力求以严谨而通俗的语言，讲述 C 语言的语法，以使初学者能够建立起正确的概念，掌握语言本身的特征；力求通过丰富的示例，讲述 C 语言的编程技术，使初学者能够学会基本的编程方法，领略到其中的真谛；力求用程序运行的结果，验证 C 语言语法的各种规则，以使初学者能够明了程序的本质，形成对 C 语言的一个整体了解。

《C 语言程序设计教程》自 2004 年 6 月问世以来，受到了许多院校师生的欢迎和好评，编者在此表示由衷的感谢！2008 年 11 月，在征集多位授课老师对本书的意见和看法后，编者对其进行了第一次修订，形成了本书的第 2 版。现在，又根据多方面的意见和建议，编者仍然在原书基本内容和结构不变的前提下，进行第二次修订，形成本书的第 3 版。这次修订的变动，归纳起来有如下几点。

（1）去除原书中残留的谬误。

（2）对原书文字做了一些修改，以减少繁杂的叙述和存在的歧义。

（3）根据一些老师的要求，本书增加了第 9 章“C 语言程序调试方法简介”，分 4 节介绍调试程序的方法：①在程序中添加调试语句，②利用编译时输出的出错信息，③监视，④断点。通过这些方法，编程者就能较为容易地发现和排除程序中的语法错误和逻辑错误，编写出较高质量和水平的 C 语言程序。

（4）为配合第 9 章，本书最后给出了附录 3：Turbo C 编译的主要错误一览。

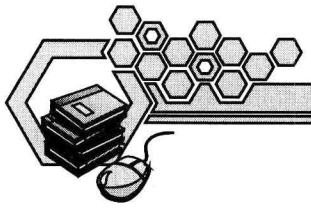
本书建议总学时数为 56，授课学时数（不含实践环节）为 44。各章的参考授课学时分配为第 1 章（3 学时），第 2 章（5 学时），第 3 章（7 学时），第 4 章（5 学时），第 5 章（7 学时），第 6 章（5 学时），第 7 章（5 学时），第 8 章（4 学时），第 9 章（3 学时）。如果安排第 9 章为学生自学，那么可把它的 3 学时分别给予第 2、3 和 8 章。

程序设计意味着实践！希望学习者能够做到 3 个“一定”：一定要以极大的热情去动手编写程序；一定要在实践过程中有百折不挠的精神；一定要深信“失败是成功之母”！只有坚持上机实践，不断体验失败，才能赢取胜利，才能到达成功的顶峰。

本书在修订过程中，得到多位同事、朋友的支持和帮助，在此不一一详列，仅致以诚挚的谢意。由于作者水平有限，书中仍然难免出现不当甚至谬误之处，恳请广大读者给予批评指正。

编　者

2012 年 5 月



目　录

第 1 章　概述 …… 1

1.1　高级语言与 C 语言 …… 1
1.1.1　程序设计语言与 C 语言 …… 1
1.1.2　简单的 C 语言程序 …… 3
1.1.3　程序设计时的算法描述 …… 5
1.2　C 语言的基本词法 …… 6
1.2.1　字符集 …… 6
1.2.2　保留字 …… 7
1.2.3　标识符及其构成规则 …… 8
1.3　Turbo C 2.0 开发环境简介 …… 8
1.3.1　主窗口的组成 …… 9
1.3.2　对源程序文件的编辑 …… 10
1.3.3　编辑的基本操作命令 …… 11
1.3.4　源程序的保存 …… 12
1.3.5　编译、连接和装配 …… 14
1.3.6　运行和观看运行结果 …… 15
习题 1 …… 16

第 2 章　数据类型、运算符与表达式 …… 18

2.1　C 语言的数据类型 …… 18
2.2　常量 …… 20
2.2.1　整型常量 …… 20
2.2.2　实型常量 …… 22
2.2.3　字符常量 …… 23
2.2.4　字符串常量 …… 24
2.3　简单变量 …… 25
2.3.1　变量的数据类型 …… 26
2.3.2　变量的存储类型 …… 27
2.3.3　变量的初始化与完整的变量说明语句 …… 29
2.3.4　变量的地址与取地址符“&” …… 31
2.4　C 语言的运算符与各种表达式 …… 32
2.4.1　算术运算符与算术表达式 …… 33
2.4.2　赋值运算符与赋值表达式 …… 35
2.4.3　关系运算符与关系表达式 …… 37
2.4.4　逻辑运算符与逻辑表达式 …… 38
2.4.5　条件运算符与条件表达式 …… 40
2.4.6　逗号运算符与逗号表达式 …… 41
2.4.7　位运算符 …… 42
2.4.8　表达式中数据类型的转换 …… 44
习题 2 …… 45

第 3 章　C 语言程序设计的 3 种基本结构 …… 47

3.1　顺序结构程序设计 …… 47
3.1.1　赋值语句、复合语句、空语句 …… 48
3.1.2　字符输入/输出函数 …… 50
3.1.3　格式输入/输出函数 …… 51
3.2　选择结构程序设计 …… 55
3.2.1　if 单分支选择语句 …… 55
3.2.2　if…else 双分支选择语句 …… 57
3.2.3　if…else if 多分支选择语句 …… 58
3.2.4　if 语句的嵌套结构 …… 60
3.2.5　switch 多分支选择语句 … 61

3.3 循环结构程序设计……66
3.3.1 while 循环语句……67
3.3.2 do…while 循环语句……69
3.3.3 for 循环语句……72
3.3.4 break 和 continue 语句……76
3.3.5 循环的嵌套结构……79
习题 3……83

第 4 章 数组……87

4.1 数组的基本概念……87
4.2 一维数组……88
4.2.1 一维数组的说明……88
4.2.2 一维数组元素的初始化……90
4.2.3 一维数组元素的引用……91
4.3 二维数组……93
4.3.1 二维数组的说明……93
4.3.2 二维数组元素的初始化……95
4.3.3 二维数组元素的引用……95
4.4 字符数组与字符串……98
4.4.1 字符数组与字符串……98
4.4.2 字符串的运算……100
4.4.3 常用的字符串处理函数……102
习题 4……106

第 5 章 指针……110

5.1 指针和指针变量……110
5.1.1 直接访问和间接访问……110
5.1.2 指针变量的说明和初始化……112
5.1.3 取地址运算符与指针运算符……114
5.2 指针与数组……119
5.2.1 指向一维数组的指针变量……119
5.2.2 指向字符串的指针变量……126
5.2.3 指向二维数组的指针变量……128
5.3 指针数组……131
5.3.1 一维指针数组的说明和初始化……132
5.3.2 指针数组元素的引用……133
习题 5……135

第 6 章 函数……139

6.1 函数的概念……139
6.1.1 函数的定义……140
6.1.2 函数的调用……142
6.1.3 函数的原型说明……146
6.1.4 变量的作用域和生命期……148
6.2 函数调用中的数据传递……152
6.2.1 参数是普通变量时的数据传递过程……152
6.2.2 参数是指针变量时的数据传递过程……154
6.2.3 参数是数组名时的数据传递过程……157
6.2.4 返回语句 return……159
6.3 指针型函数……161
6.3.1 指针型函数的定义方法……161
6.3.2 指针型函数的使用……161
习题 6……163

第 7 章 用户自定义的数据类型……167

7.1 结构型数据类型……167
7.1.1 结构型数据类型的定义……168
7.1.2 结构类型变量的说明与初始化……169
7.1.3 结构变量成员的引用……171
7.1.4 结构数组的说明与初始化……173
7.2 指向结构类型的指针……176
7.2.1 指向结构类型变量的指针……176

7.2.2 指向结构类型数组的指针 …… 178
7.2.3 C 语言的内存管理函数 …… 179
7.2.4 自引用结构类型和链表 …… 184
7.3 共享型数据类型 …… 189
7.3.1 共享型数据类型的定义 …… 189
7.3.2 共享类型变量的说明和使用 …… 189
7.4 枚举型数据类型 …… 192
7.4.1 枚举型数据类型的定义 …… 192
7.4.2 枚举类型的使用 …… 193
7.5 预处理和起别名 …… 195
7.5.1 宏命令 #define …… 196
7.5.2 文件包含命令#include … 199
7.5.3 起别名语句 typedef …… 199
习题 7 …… 201

第 8 章 C 语言的文件操作函数 …… 206

8.1 文件及文件型指针 …… 206
8.1.1 C 语言的文件概念 …… 206
8.1.2 C 语言的文件结构类型及其指针 …… 208
8.2 文件的打开与关闭函数 …… 209
8.2.1 文件打开函数：fopen() …… 209
8.2.2 文件关闭函数：fclose() …… 211
8.2.3 标准设备文件的使用 …… 213
8.3 文件的读/写操作 …… 213
8.3.1 文件尾测试函数 …… 213
8.3.2 读/写字符函数 …… 214
8.3.3 读/写字符串函数 …… 217
8.3.4 读/写数据函数 …… 221
8.3.5 格式读/写函数 …… 223
8.4 文件操作中的其他函数 …… 226
8.4.1 文件头定位函数 …… 226
8.4.2 文件随机定位函数 …… 228
8.4.3 错误测试函数 …… 230
习题 8 …… 231

第 9 章 C 语言程序调试方法简介 …… 235

9.1 在程序中添加调试语句 …… 235
9.2 利用编译时输出的出错信息 …… 238
9.3 监视 …… 244
9.3.1 C 语言提供的监视命令 …… 245
9.3.2 监视调试举例 …… 247
9.4 断点 …… 252
9.4.1 C 语言提供的断点命令 …… 253
9.4.2 利用断点调试举例 …… 253

附录 1 常用的 Turbo C 库函数 …… 257

附录 2 常用字符的 ASCII 码 …… 260

附录 3 Turbo C 编译的主要错误一览 …… 261

参考文献 …… 268

第1章 概述

“由表及里”是一种认识问题的方法。这一章就先从外表来看一下 C 语言，以便能在人们的脑海里对它形成一个初步的印象：什么是计算机程序？什么是计算机的程序设计语言？有哪几种程序设计语言？C 语言在其中处于什么位置？用 C 语言编写的程序大致是个什么样子？在什么环境里能够编写 C 语言的程序？有了这些初步的印象，读者就会知道应该如何去学习 C 语言，知道在学习过程中把自己的注意力集中在什么地方。这一切，对于学习、掌握 C 语言，肯定是至关重要的。

本章着重讲述以下 4 个方面的内容：

（1）C 语言程序的基本组成；

（2）C 语言的基本词法（字符集、保留字和标识符的构成）；

（3）用 C 语言编写程序时的 4 项工作；

（4）Turbo C 开发环境简介。

1.1 高级语言与C语言

1.1.1 程序设计语言与C语言

按照字典的说法，“程序”是指一件事情进行的先后次序，“语言”则是用来表达意思、交流思想的工具。因此，“计算机程序”即是要让计算机去完成的事情的先后次序。要让计算机去完成的事情，当然是由人提供给计算机去做的。这就是说，人要使用一种办法，把自己要计算机去做的事情描述出来，然后才能提交给它去做。于是，通常就把人与计算机之间“表达意思、交流思想”的那种工具，称为“计算机程序设计语言”。人们就是用计算机程序设计语言来编写计算机程序，然后交于计算机去执行的。

自世界上第一台计算机在 1946 年问世以来，用于编写计算机程序的程序设计语言，由所谓的机器语言发展到汇编语言，又由汇编语言发展到高级语言。

计算机进行计算和信息处理，是通过执行一条一条指令来完成的。指令是让计算机执行各种操作的命令，每条指令由包括“操作码”和“地址码”两部分的二进制代码（一串 1 和 0）组成。操作码规定计算机要做的运算；地址码告诉计算机由哪些数来参加运算，在什么地方能找到它们，计算完毕后的结果应该存放到哪里等。一台计算机所有指令的集合，称为该机器的“指令系统”。在计算机刚出现的年代，人们是直接用计算机指令来编写计算机程序，完成人与计算机之间的“思想”交流的。因此，常称计算机的指令系统为计算机的“机器语言”。用机器语言编写程序与计算机“交谈”时，计算机不必通过任何就翻译处理能够立即理解和执行。因此，那样的程序具有质量高、执行速度快和占用存储空间少等优点。但是，由于每条指令都是长长的一串 1 和 0（二进制代码），很显然用它来编写程序，非常缺乏直观性，难学、难记、难检查以及难修改。

为了克服机器语言的缺点，出现了汇编语言。汇编语言用每个操作相应的英文单词缩写代替指令中的二进制操作码，用各种符号或数字代替指令中的地址码。例如，用 MOV 表示数据传送操作，用 ADD 表示加法操作等。在汇编语言中用便于记忆的符号（称为“助忆符”）来代替机器指令中的操作码，用各种符号或数字代替指令中的地址码，使得机器语言得以“符号化”。比起机器语言来，它显然好记了，读起来容易了，检查、修改也方便了。但是这样一来，用汇编语言编写的程序，计算机却不能直接理解和接受，它必须要由一个起翻译作用的程序将其翻译成机器语言程序，然后才能去执行。这个起翻译作用的程序，通常被称为“汇编程序”，这个翻译过程，称之为“汇编”。

由于汇编语言的指令基本上与机器指令一一对应，因此它是依赖于具体机器的一种语言，每台计算机的汇编语言指令集合都是不同的。通常，称具有这种特性的语言为“面向机器”的程序设计语言。汇编语言的缺点是不具有通用性和可移植性，它与人们习惯使用的自然语言和数学语言相差甚远，因此又出现了所谓的高级语言。

高级语言是一种很接近于人们习惯使用的自然语言（即人们日常使用的语言）和数学语言的程序设计语言。在用高级语言编写计算机程序时，允许出现规定的英文词汇（如第 1.2.2 节中的保留字）；在书写计算式时，所用的运算符号和组成的算式，与我们日常用的数学式子差不多（比如第 2 章中的运算符和不等式）。因此，人们用它来编写计算机程序，比起使用机器语言和汇编语言，显然要方便得多。

用高级语言编写的程序，称为“源程序”。如同用汇编语言编写的程序计算机不能直接理解和执行一样，用高级语言编写的程序，计算机也不能直接理解和执行，也必须要有一个“翻译”过程，先把源程序翻译成机器语言的程序，然后再让计算机去执行这个机器语言程序。对于高级语言来说，翻译过程有两种方式：一是事先编好一个称为“编译程序”的机器指令程序，它把源程序整个地翻译成用机器指令表示的机器语言程序（这个翻译结果通常称为“目标程序”），然后执行该目标程序，这种翻译过程称为“编译”，如图 1-1（a）所示。另一个是事先编好一个称为“解释程序”的机器指令程序，它把源程序逐句翻译，翻译一句执行一句，这种翻译过程称为“解释”，如图 1-1（b）所示。

C 语言是一种高级语言，它用比较接近人的思维和表达方式来描述问题、编写计算机程序，然后以编译的方式进行翻译。

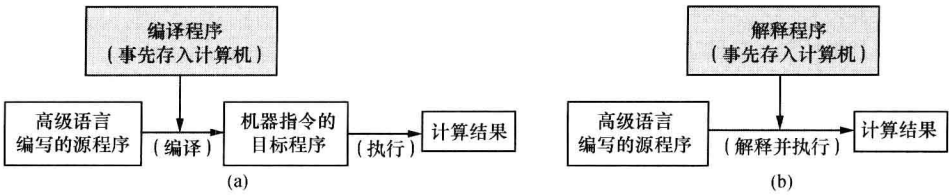


图 1-1　编译和解释两种翻译方式示意图

例 1-1　分别用机器语言、汇编语言和 C 语言描述计算式：z=x+y。

解　假定事先把数 235 和 368 分别存放在地址为 0110 及 0120 的两个存储单元中，那么描述它们相加的指令序列，对于机器语言可以有如下程序段：

```
A11001
03062001
A33001
```

三条指令的含义是：把 0110 中存的数取到寄存器 AX；然后把 0120 中的数取出，与 AX 里的内容相加，结果在 AX 中；最后把 AX 里的结果存到 0130 地址单元。

在相同假定下，对于汇编语言可以有如下程序段：

```
MOV AX, [0110]
ADD AX, [0120]
MOV [0130], AX
```

对于 C 语言，无须有什么假定，而是直接编写程序段：

```
int x=235, y=368;
z=x+y;
```

它表示让变量 x 和 y 分别取值 235 和 368，求和后把结果存放在变量 z 里。编程者并不需要去管数据放在何处，就像是写一道算术题似的。

由此例可以看出，机器语言程序完全没有直观性可言，如果不了解机器指令 A1 表示的是将跟随其后单元中的内容送至寄存器 AX，那么根本无法知道它的含义。对于汇编语言，MOV 是英文 move 的简写，因此可以知道它是要把一个数据送到寄存器 AX 中去。可见，汇编语言具有一定的直观性，便于人们记忆。再看 C 语言，它简直就是使用人们习惯的数学表达式来描述加法。可见，学习用 C 语言来编写计算机程序，人们容易接受。

1.1.2　简单的 C 语言程序

先让我们通过两个简单的 C 语言程序，大致领略一下用 C 语言来编写程序时的通常做法，从而归纳出 C 语言程序的一些特点。

例 1-2　用 C 语言编写一个程序，它接收从键盘输入的两个整数，求和后打印输出。

解　（1）程序实现。

```
#include "stdio.h"
main()
{
  int m, n, sum;                    /* 变量说明 */
  scanf ("%d%d", &m, &n);           /* 从键盘输入数据 */
  sum=m+n;                          /* 求和 */
  printf ("sum=%d\n", sum);         /* 打印输出 */
```

```
    getchar();
}
```

（2）分析与讨论。

在 C 语言中，以符号“/*”开始、“*/”结束的中间部分，是对左边程序语句的注释。不难通过上面程序中给出的注释，读懂花括号里整个程序 5 条语句的意思。

第 1 条语句“int m, n, sum;”，表示 m、n 和 sum 是 3 个变量，前面的 int 说明它们都是整数型的（即单词 integer 的简写）。

第 2 条语句“scanf ("%d%d", &m, &n);”是一条格式输入语句，其中的&m 和&n 表示变量 m 和 n 所对应的内存单元地址。该语句的功能是按照格式符“%d”的规定，从键盘接收两个十进制的输入数据（格式符%d 中的字母 d 限定输入的数据是十进制的），分别存放到地址&m 和&n 指定的存储单元中。

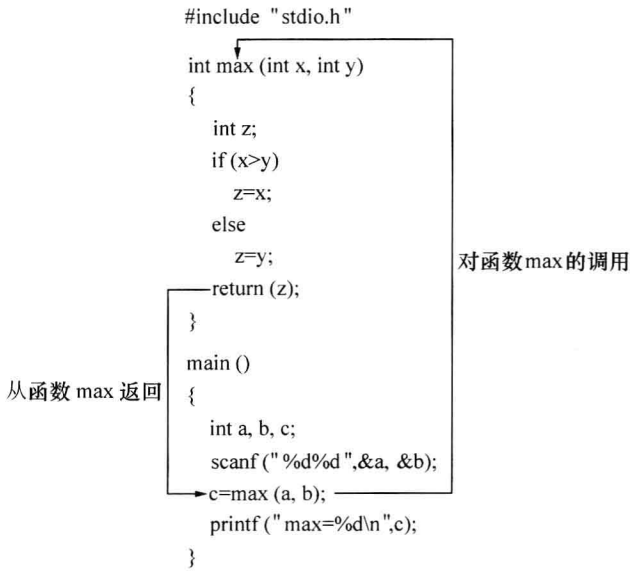
第 3 条语句“sum=m+n;”的含义是，把变量 m 和 n 里的数据相加，然后将结果存入变量 sum 保存。注意，C 语言中的符号“=”，不是等号，而是赋值运算符，表示是把右端的计算结果送给左端变量的一个操作过程。

第 4 条语句“printf ("sum=%d\n", sum);”，是一条格式打印输出语句，表示将变量 sum 的当前值，按照格式符“%d”的规定输出一个十进制整数。比如说，如果现在从键盘上输入的两个数是 3 和 5，那么，在显示器上就应该输出信息：sum=8。

第 5 条语句“getchar();”，使程序处于等待状态，以便向用户提供观看运行结果的机会。只有在用户从键盘上输入任何一个字符后，才结束等待返回程序。

例 1-3 用 C 语言编写程序，它接收从键盘输入的两个整数，将其中的大数打印输出。

解 （1）程序实现。



（2）分析与讨论。

日常生活中，人们总是把大的、复杂的事情，化为若干小的、简单的事情去处理。在程序设计时，也常采用这种方法。在这个程序里，我们把接收键盘的输入和打印输出作为一件事情来处理（表现在 main()里），把判断两个数的大小作为另一件事情来处理（表现在 max()里）。然后通

过一定的办法，把这两件事情拼接到一起，整个事情也就完成了（注意，程序中的箭头线是用来表明函数间的调用关系的）。

该程序由两个函数组成：一个名为 main，一个名为 max。在 main 里，使用格式输入语句 scanf 往变量 a 和 b 里输入数据，然后用 a 和 b 去调用 max。max 的功能是比较两个数的大小，把大数存入变量 z，通过 return 语句，把 z 的值返回。这样，从 max 返回时，就把 z 中的结果送给 main 里的变量 c。最后，由格式打印语句 printf，把 c 的内容打印出来。

由以上两个例子，可以初步归纳出用 C 语言编写计算机程序时，具有如下特点。

（1）C 语言程序是由一个一个函数组成的，函数是 C 语言程序的基本单位。比如，例 1-2 的程序，是由一个名为 main 的函数组成的；例 1-3 的程序，是由一个名为 main 的函数和一个名为 max 的函数组成的。

（2）每一个 C 语言程序，都有一个，且只有一个名为 main 的主函数，整个程序从它开始执行。至于 main 函数在整个程序中所放的位置，与它作为程序开始执行的地位没有什么关系。也就是说，main 函数可以安排在整个程序的最前面、中间或后面。

（3）C 语言程序中的每一个语句，都以分号作为自己的结束。也就是说，在 C 语言中，分号“;”是一个语句的结束标志。

（4）在 C 语言程序中，可以用/*……*/形成注释，以对程序中的所需部分做出说明。/*是注释的开始符，*/是注释的结束符，必须配对使用。

1.1.3　程序设计时的算法描述

用计算机程序设计语言编写程序，首先应该选定要用的计算公式，制定解决问题的步骤，确定程序采用的结构（到第 3 章时会知道，程序的结构主要有 3 种形式：顺序结构、选择结构以及循环结构）等，然后才能真正动手去编写程序和上机调试。这个在真正动手之前的准备环节，就是所谓的算法描述阶段。这个阶段，对于问题的解决，无疑是非常重要的。

为了把解决问题的方法和步骤（也就是所谓的算法）描述出来，可以借助于人们日常使用的语言（称为“自然语言”）；可以借助于传统的流程图；可以借助于所谓的 N-S 流程图；也可以借助于介于自然语言和计算机语言间的文字和符号（称为“伪代码”）。总之，描述的方法是多样的，目的只有一个，即按照算法的描述编写程序时，思路会更加清晰。

本书列举的程序都是比较简单的，因此不去专门关注算法的描述。不过为了帮助读者对所编程序的理解，有时会给出程序的流程框图。图 1-2 给出了画流程框图时常用的一些符号。

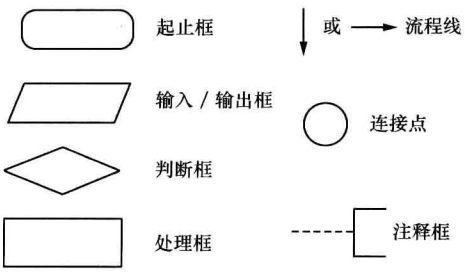


图 1-2　常用的流程图符号

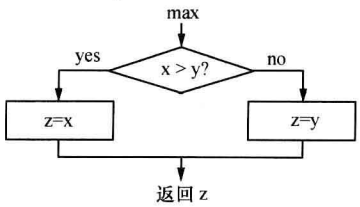


图 1-3　函数 max 的流程图

比如可利用这些符号，为例 1-3 中“判断两个数大小”的函数 max 绘制流程图，如图 1-3 所示。图中清楚地反映出“x>y?”是一个条件，如果条件成立（yes），那就去做“z=x”；否则（no）就去做“z=y”。这是一种根据条件做出选择的流程图，它有两个出口：yes 和 no。

1.2 C 语言的基本词法

任何一种语言，都有自己的单字、单词和语句的构成规则。学会了这些知识，才能用它们书写出精彩的文章。C 语言作为计算机的一种程序设计语言，当然也有它自己允许使用的字符集、基本词类（即保留字），也有书写时的各种规则和语法。只有学习、遵从它们，才能编写出符合要求的各种精彩程序来。

1.2.1 字符集

允许出现在 C 语言源程序中的所有字符的总体，称为 C 语言的“字符集”。它由数字、英文字母、图形符号以及转义字符 4 部分组成。

（1）数字：10 个十进制的数字，即 1，2，3，4，5，6，7，8，9，0。

（2）英文字母：26 个大写英文字母 A～Z，26 个小写英文字母 a～z。

（3）图形符号：表 1-1 列出了 C 语言允许使用的图形符号。

表 1-1　　C 语言图形符号表

～	波浪号	)	右圆括号	:	冒号
`	重音号	_	下划线号	;	分号
!	惊叹号	–	减号	"	双引号
@	A 圈号	+	加号	'	单引号
#	井号	=	等号	<	小于号
$	美元号	|	或符号	>	大于号
%	百分号	\	反斜杠号	,	逗号
^	异或号	{	左花括号	.	句号
&	与符号	}	右花括号	?	问号
*	星号	[	左方括号	/	正斜杠号
(	左圆括号	]	右方括号		空格

（4）转义字符：在 C 语言源程序中，可以用在反斜杠号（\）后面跟随特定的单个字符或若干个字符的方法，来表示键盘上的字符以及某些不可见的功能控制符（如退格、换行等）。这时，尾随反斜杠后的字符就失去了原有的含义，而赋予了新的特殊含义。通常称反斜杠号为转义符，称反斜杠以及随后的字符整体为一个“转义字符”。表 1-2 是 C 语言的转义字符表。

注意：只有把转义符（反斜杠）放在表 1-2 中所列出的字符前面时，才能构成转义字符，否则不起任何作用。比如\w，由于反斜杠后面跟随的字符 w 不在表 1-2 中，所以不构成转义字符，它被视为是小写字母 w。

例 1-4　区别“n”和“\n”。

解　当程序中出现“n”时，代表的是英文中的一个小写字母，比如例 1-2 里的变量 n；当程序中出现“\n”时，反斜杠后跟随的 n 就不再是英文中的小写字母 n，这个整体被视为是回车换行符。所以，在例 1-2 或例 1-3 的 printf()中的“\n”，表示回车换行符。

表 1-2　　C 语言的转义字符表

转义字符	含　义	转义字符	含　义
\n	回车换行符	\a	响铃符号
\t	Tab 符号	\"	双引号
\v	垂直制表符号	\'	单引号
\b	左退一格符号	\\	反斜杠
\r	回车符号	\ddd	1～3 位 8 进制数 ddd 对应的键盘符号
\f	换页符号	\xhh	1～2 位 16 进制数 hh 对应的键盘符号

例 1-5　在 C 语言程序中写“\101”、“\x41”，它们分别表示什么意思？

解　按照表 1-2 的规定，在反斜杠后跟随 1～3 位数时，就把这些数字理解为是某个键盘符号所对应的八进制 ASCII 码值。101 这个八进制数相当于十进制数 65，查书后的附录 2，知道是大写字母“A”。所以，“\101”就是大写的英文字母“A”。类似地，应该把“\x41”里的 41 视为键盘符号对应的十六进制 ASCII 码值。因此，它也是大写的英文字母“A”。

注意：“\xhh”中的字符“x”，只起到一个标识后面的数是十六进制的作用，没有别的含义。由例 1-5 可知，转义字符“\ddd”、“\xhh”，向用户提供了一种用八进制或十六进制的 ASCII 码值来表示各个字符的方法。

1.2.2　保留字

在 C 语言中，具有特定含义的、用于构成语句成分或作为存储类型和数据类型说明的那些单词，被统称为“保留字”，也称“关键字”。C 语言的保留字只能小写。表 1-3 列出了 C 语言中可使用的所有保留字。随着学习的深入，这些保留字我们基本上都会遇到的。

表 1-3　　C 语言的保留字表

保留字	含　义	保留字	含　义	保留字	含　义
char	字符型	void	空值型	while	当
int	整型	const	常量型	do	做
long	长整型	volatile	可变量型	break	终止
short	短整型	auto	自动	continue	继续
float	单精度实型	extern	外部	goto	转向
double	双精度实型	static	静态	return	返回
unsigned	无符号型	register	寄存器	switch	开关
signed	有符号型	typedef	类型定义	default	默认

续表

保 留 字	含 义	保 留 字	含 义	保 留 字	含 义
struct	结构式	if	如果	case	情况
union	共用式	else	否则	sizeof	计算字节数
enum	枚举式	for	对于		

1.2.3 标识符及其构成规则

在 C 语言中，用户为了区分程序中出现的常量、变量、函数和数组等，就给它们取不同的名字。组成名字的字符序列，称为“标识符”。因此，标识符是用户给程序中需要辨认的对象所起的名字。一个标识符必须符合下面所列的语法规则。

（1）标识符只能以字母或下画线开头。

（2）在第一个符号的后面，可以跟随字母、数字或下划线。

（3）标识符中区分字母的大、小写。

（4）标识符的长度一般不超过 8 个字符。

（5）C 语言的保留字不能作为标识符使用。

例 1-6 试判断下面所给出的字符序列，哪一个是正确的 C 语言标识符。

```
x          _906      A203      aBBC      C.508      int
y-56       gb?       b_B64     2abc      ABBC
```

解 根据构成标识符的语法规则可知，上述字符序列里，正确的标识符是：

```
x          _906      A203      aBBC      b_B64      ABBC
```

不正确的标识符是：

C.508（句号不允许出现在标识符里）　　y-56（减号不允许出现在标识符里）

gb?（问号不允许出现在标识符里）　　2abc（标识符不允许以数字开头）

int（保留字不允许作为标识符）

注意：由于 C 语言区分大、小写，所以 aBBC 和 ABBC 是两个不同的标识符。

1.3 Turbo C 2.0 开发环境简介

真正要运行一个用 C 语言编写的程序，至少要做如下的 4 项工作。

（1）编辑。通过使用编辑器，把 C 语言程序录入计算机，并以文件的形式存放到磁盘上，这个过程称为“编辑”。它将产生出以“.C”为扩展名的源程序文件。

（2）编译。源程序不能直接执行，必须通过 C 语言的编译程序将它“翻译”成由机器指令组成的目标程序，这个过程称为“编译”。它产生出以“.OBJ”为扩展名的目标程序文件。

（3）连接装配。目标程序仍不能立即在机器上执行，因为程序中还可能会用到 C 语言自身提供的系统库函数，例如 printf()、scanf()等，需要把它们与产生的目标程序连接在一起，形成一个整体，这个过程称为“连接装配”。它将产生出以“.EXE”为扩展名的可执行程序文件。

（4）执行。运行可执行文件，以获取所需要的结果。

1.3.1　主窗口的组成

现在，我们很容易从网上下载 C 语言的各种编译程序，它们的使用方法基本上是相同的。本书采用的是早期 Borland 公司的 Turbo C 2.0 版本。在计算机上安装好 Turbo C 2.0 后，在 DOS 提示符下，键入“TC”，即可进入到它的主窗口，如图 1-4 所示。

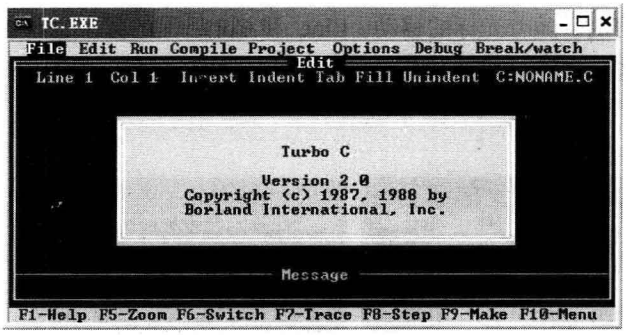


图 1-4　Turbo C 的主窗口

Turbo C 2.0 向使用者提供的是一个集成开发环境。也就是说，在 Turbo C 2.0 所提供的平台上，用户可以完成编辑、编译、连接装配以及运行的全部工作。下面，对 Turbo C 2.0 的使用做一个简要的介绍。

主窗口正中给出的是 Turbo C 的版本信息，只要用户在键盘上按任意一个键，版本信息就会立即消失。主窗口由主菜单、编辑区、信息区和功能键提示行 4 个部分组成。

1．主菜单

位于主窗口标题栏（有 TC.EXE 字样）下面的是 Turbo C 的主菜单，它有 8 个菜单项：File（文件）、Edit（编辑）、Run（运行）、Compile（编译）、Project（项目）、Options（选项）、Debug（调试）和 Break/watch（断点/监视）。除 Edit 外，其余每个主菜单项都还有下拉子菜单，用以实现各种操作。后面会对 File、Run 和 Compile 三者的使用做一些介绍。

2．编辑区

位于主菜单下面标有 Edit 字样的区域是 Turbo C 的程序编辑区，用于输入新的 C 语言源程序，或对已有的源程序进行编辑。

3．信息区

标有 Message 字样的区域是 Turbo C 的信息区，用于显示编译和连接时的有关信息（比如成功与否，出错信息等）。

4．功能键提示行

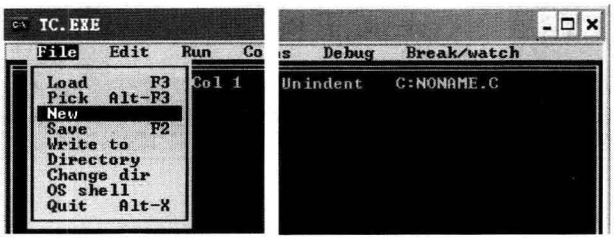
位于屏幕最下方，给出常用的 7 个功能键，它们是 F1（帮助）、F5（分区控制）、F6（转换）、F7（跟踪）、F8（单步执行）、F9（生成目标文件）和 F10（菜单）。对于初学者，应该知道在编辑完一个程序后，可以按 F9 键，它能进行编译和连接，产生出“.OBJ”文件以及“.EXE”文件，但是不运行。初学者也应该知道，在非菜单状态下（比如处于编辑状态时），按 F10 键就可以进到主菜单，这时通过按“←”、“→”键可以选择主菜单项，在主菜单项的下拉子菜单中，通过按“↑”、“↓”键可以选择子菜单项。

1.3.2 对源程序文件的编辑

如果要建立一个新的 C 语言源程序，应该从主菜单项 File 的下拉菜单中选择 New，如图 1-5（a）所示。按回车键后，整个编辑区被清空，光标定位在该区左上角（第 1 行，第 1 列）。这样，用户就可以输入和编辑源程序了。编辑区最前面（即首行）有一行文字：

```
Line 1  Col 1  Insert  Indent  Tab  Fill  Unindent  C:NONAME.C
```

在输入源程序内容时，随着光标的移动，Line 和 Col 后面的数字也跟着改变，以标明输入光标目前所在的位置。这行的最右端给出了现在新编辑文件的默认名：C:NONAME.C，如图 1-5（b）所示。



(a) (b)

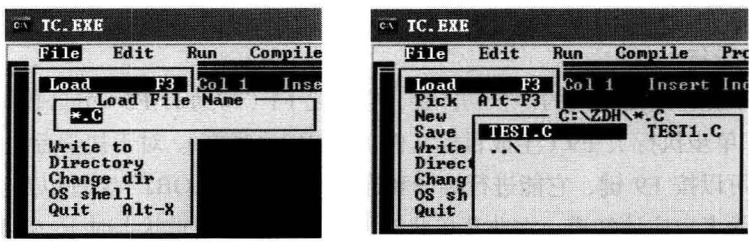
图 1-5 编辑新文件命令 New 和默认文件名

如果要对一个已经存在的 C 语言源程序进行编辑，那么应该从主菜单项 File 的下拉菜单中选择 Load，如图 1-6（a）所示。在选中 Load 之后，屏幕上会弹出一个包含“*.C”的“Load File Name（装入文件名）”对话框。此时，可以有两种操作方式，一是直接输入所需要的文件路径和文件名，这样，Turbo C 就会按照输入的信息，找到该文件，把它装入内存，在编辑区里显示出来，编辑区首行右端将由该文件的名字取代“C:NONAME.C”。另一个是删除“*.C”，键入路径名后按“Enter”键，这样，Turbo C 就会把指定路径（目录）下所有的 C 语言源程序文件显示出来，供用户选择，如图 1-6（b）所示。在图 1-6（b）中可以看出，当前目录是：

```
C: \ ZDH \
```

显示出的“TEST.C”、“TEST1.C”，是该目录下的 C 源程序文件。

如果按第 1 种方式操作时，Turbo C 在所给路径上找不到用户键入的文件名，这意味该文件不存在。于是，Turbo C 会自动为用户建立这个文件，清空编辑区，等待用户输入新的源程序。所以，这时该操作相当于创建一个新文件，功能与在 File 下选择 New 命令相同。



(a) (b)

图 1-6 编辑已有文件命令 Load